



l e a n  
software development

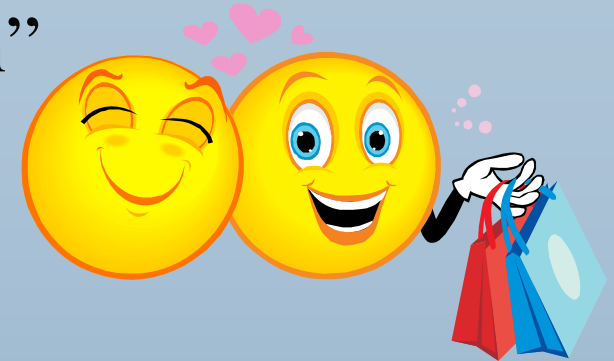
# Lean Software Development

*Faster – Better – Cheaper*

# Characteristics of “Lean” Companies:



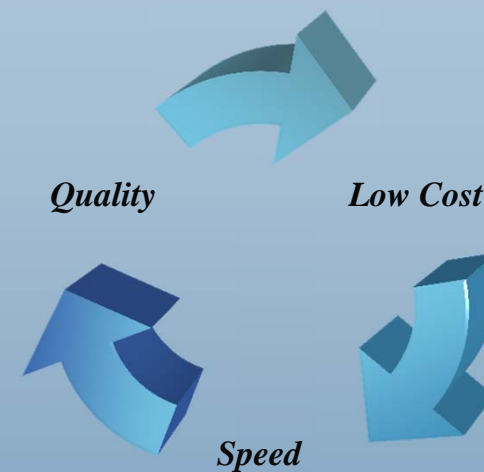
1. They don't call themselves “Lean”
  - “The ----- Way”
2. Customer Obsession
  - Stay Profitable to Stay in Business
3. Operational Discipline
  - Things. Just. Work.
4. Highly Engaged People
  - The Law of Reciprocity
5. System-Level Incentives
  - Long Term Thinking
6. Highly Successful Business



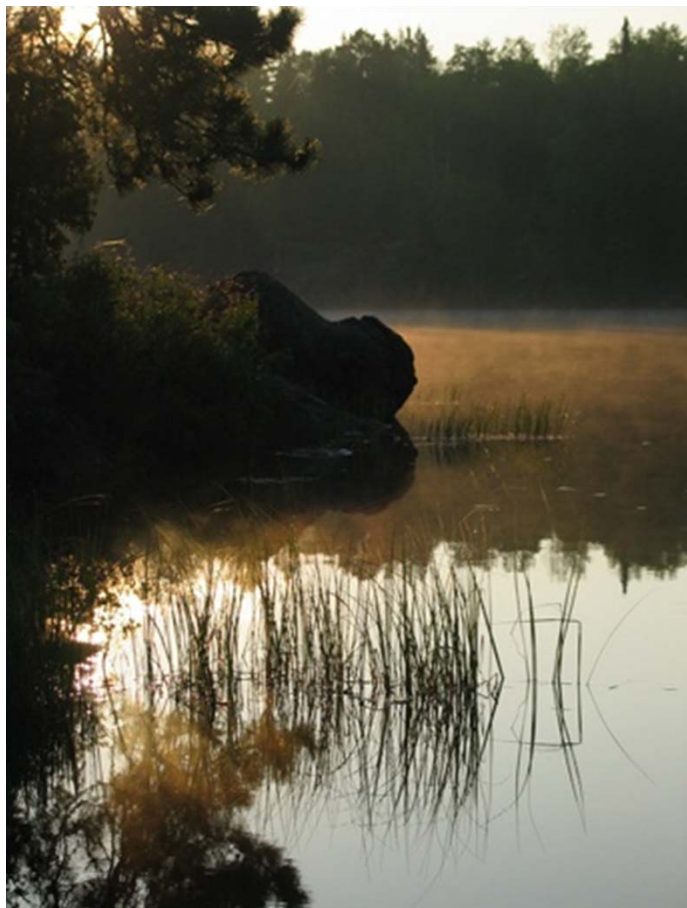
# *Principles of Lean Product Development*



1. Optimize the Whole
2. Eliminate Waste
3. Build Quality In
4. Learn Constantly
5. Deliver Fast
6. Keep Getting Better
7. Engage Everyone



l e a n



# lean

software development

## The Impact of the Pace of Delivery

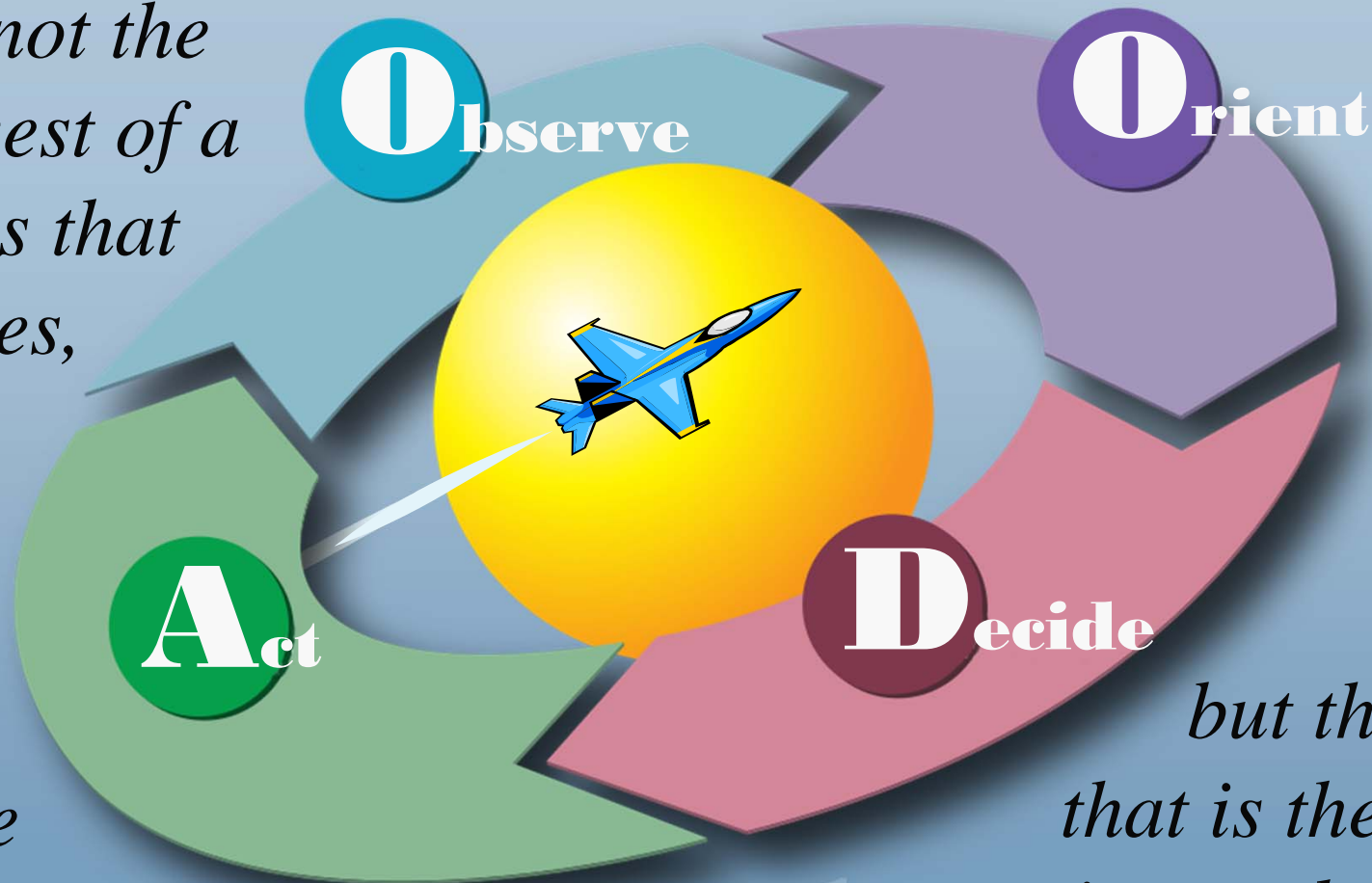
*The Fastest Learner Wins*

# *The Fastest Learner Wins*

OODA Loop: Developed by USAF strategist John Boyd



*“It is not the  
strongest of a  
species that  
survives,*



*nor the  
most intelligent,*

*but the one  
that is the most  
responsive to change.”*

– Charles Darwin

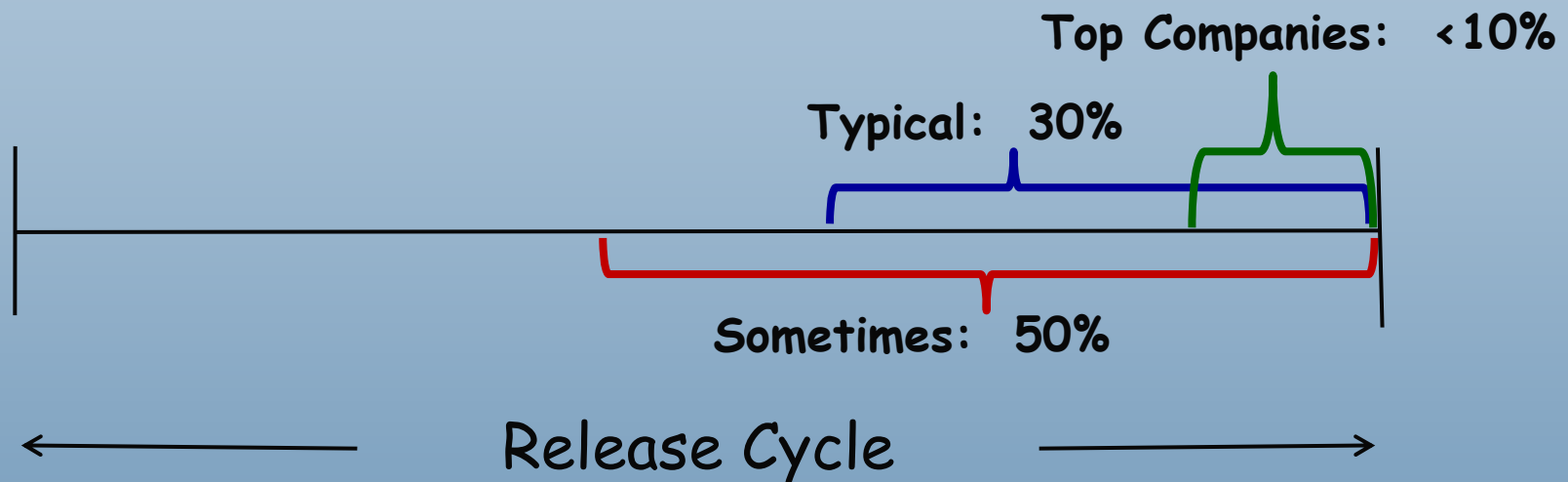
# *Release Cycle* *≥6 Months*



## *Development Cycle*

When in your release cycle do you try to freeze code and test the system?

What percent of the release cycle remains for this “hardening”?

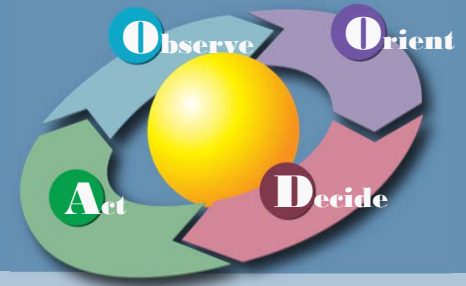


## *Governance Model*

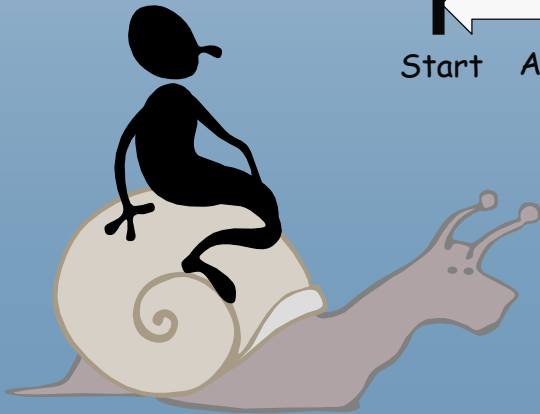
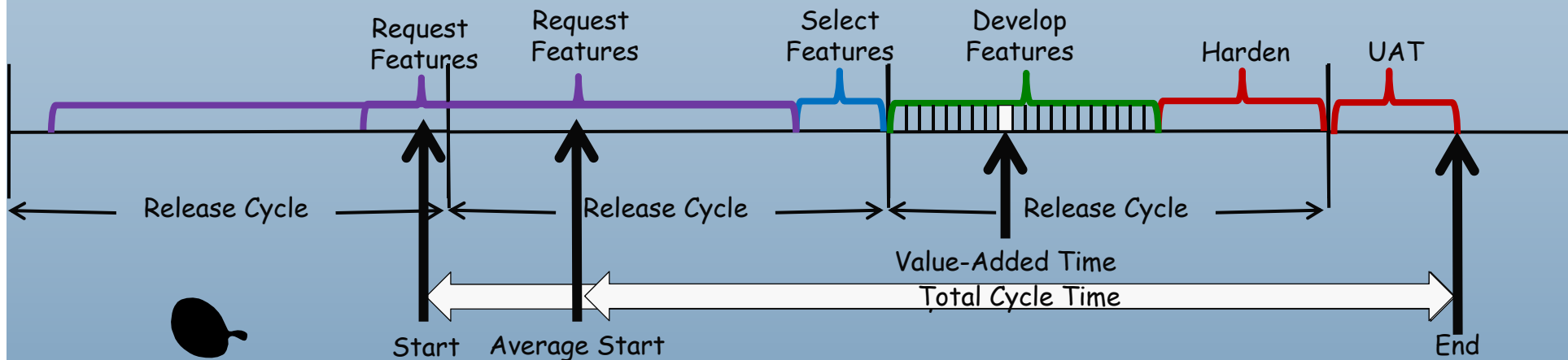
When in your release cycle do you try to choose the features for a release?



# Release Cycle ≥ 6 Months



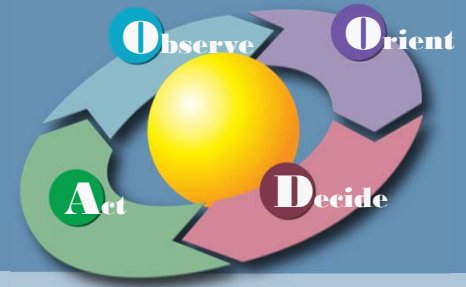
## Quick & Dirty Value Stream Map:



## Business Model:

- × Software installed at customer site
- × Support each release
- × Avoid releases

# *Release Cycle Quarterly*

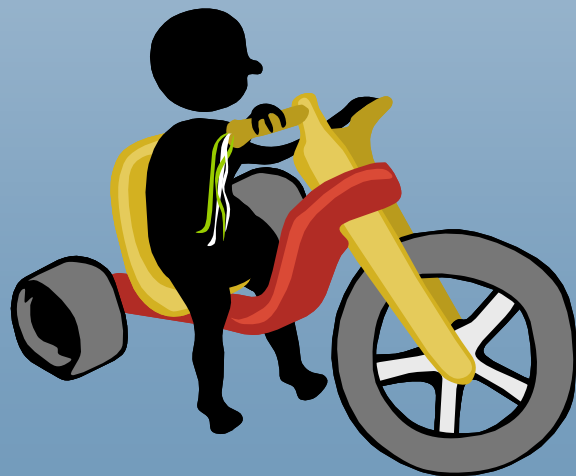


Hardening must be  $\leq 2$  weeks.

Typically: 2-4 week iterations

Code from each iteration goes to integration testing

Automated integration testing becomes necessary



The Big Bang becomes obsolete



# *Specification by Example*



For each feature:

## 1. Design

- a. Specify: Discuss and agree on examples of intended behavior.
- b. Automate: Put the examples in a framework such as cucumber.



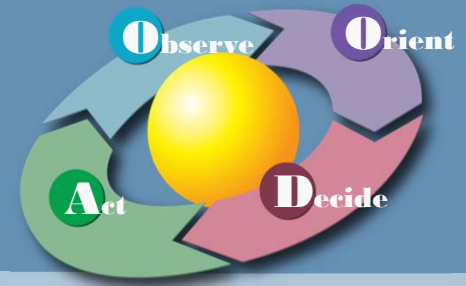
## 2. Implement

- a. Develop: Write feature code using TDD.
- b. Refactor: Clean up the code to keep it simple.
- c. Regression: Check that all completed examples work.

## 3. Deploy

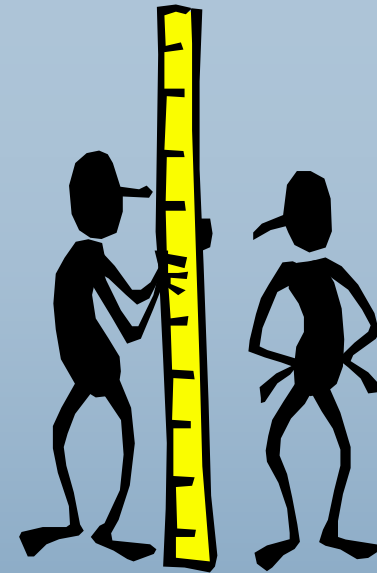
- a. Feedback: Determine value to stakeholders ASAP.

# Release Cycle Quarterly



## Business issues:

- ✓ How to price and sell releases?
- ✓ Which releases to support?
  - ✗ Supporting multiple branches can create a support nightmare
- ✓ Public vs. Private releases?



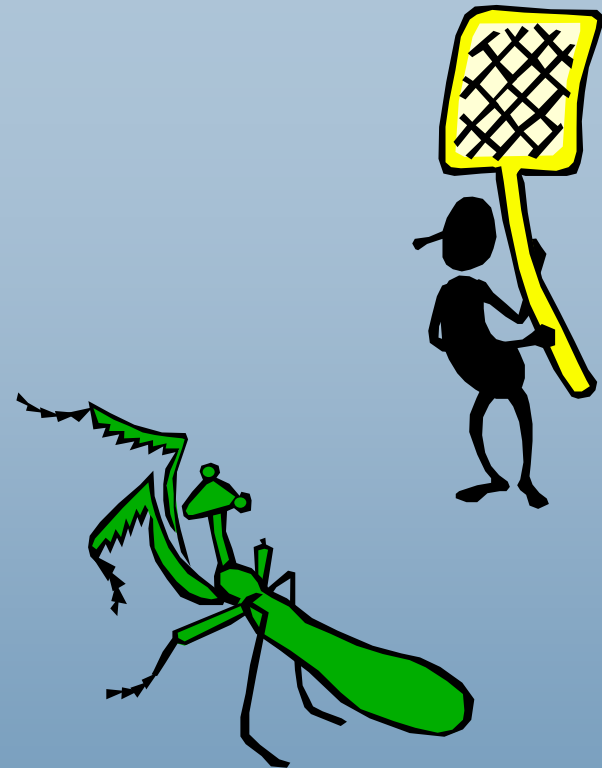
## Governance:

- ✓ Teams stay with a system – there isn't time to change
- ✓ There isn't time for variance-based (project) metrics
- ✓ Success is measured based on customer satisfaction, revenue improvement, etc.

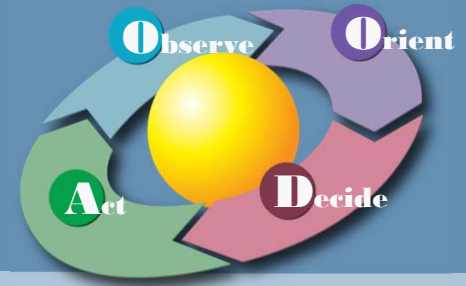
# *Basic Disciplines*



1. Low Dependency Architecture
2. Coding Standards
3. Source Control / Configuration Mgmt
4. Automated Specification Examples
5. Automated Unit Tests
6. **STOP** if the tests don't pass
7. Design/Code Reviews
8. Refactoring is a Habit
9. Continuous Integration
10. System Testing / UAT – early & often
11. Customer Validation
12. Escaped Defect Root Cause Analysis



# Release Cycle Monthly



Now you need:

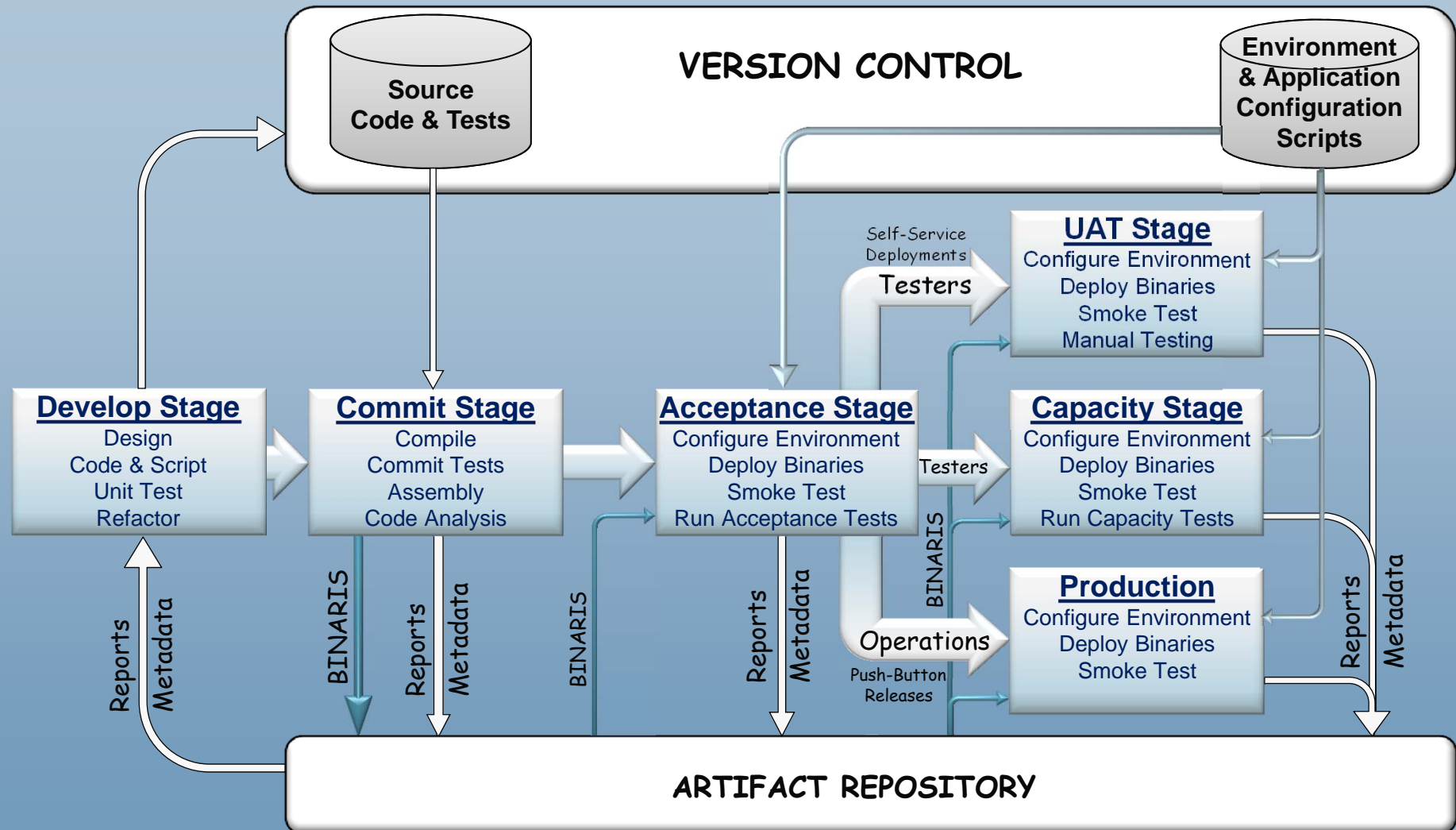
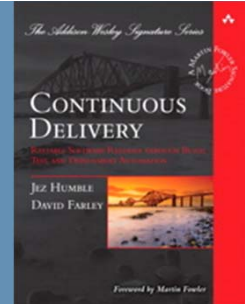
- ✓ Cross Functional Team
- ✓ Visualization
- ✓ Short Daily Meetings
- ✓ SBE/TDD working!
- ✓ Hardening  $\leq 3$  days



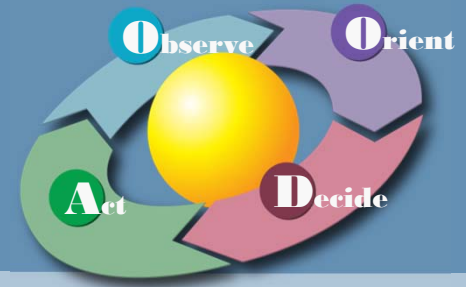
Business Environment  
Works best for:

- ✓ Software as a Service (SaaS)
- ✓ Internal Software

# A Typical Deployment Pipeline



# *Release Cycle* *Weekly/Daily/Continuous*



Kanban works well

Iterations become irrelevant

High discipline is fundamental

Estimating is largely unnecessary

Rapid cycles of learning drive portfolio decisions

The team is everyone.



DevOps:

Test & deployment automation is essential

Business Issues:

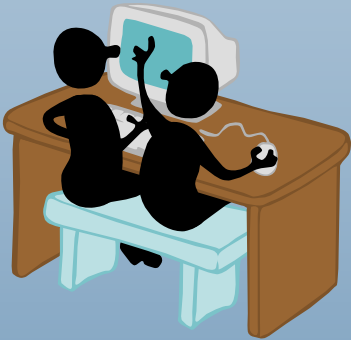
Increasingly common in startups



# The Lean Startup

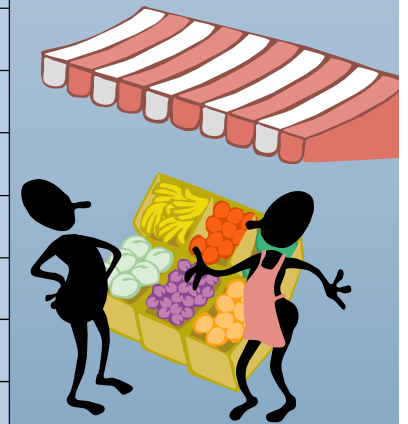
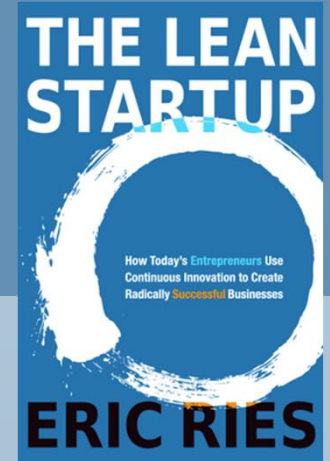
## Agile Vs. Lean Startup

Adapted from similar  
chart posted by Joshua  
Kerievsky, Industrial  
Logic Blog† August, 2011



†<https://elearning.industriallogic.com/gh/submit?Action=PageAction&album=blog2009&path=blog2009/2011/agileVsLeanStartup&devLanguage=Java>

Agile	Lean Startup
Product Roadmap	Business Model Canvas
Product Vision	Product Market Fit
Release Plan	Minimal Viable Product
On-Site Customer	“Get Out Of The Building”
Iteration	Build-Measure-Learn Loop
Iteration Review	Persevere or Pivot
Backlog	“To Learn” List
User Story	Hypothesis
Acceptance Test	Split Test
Definition of Done	Validated Learning
Continuous Integration	Continuous Deployment
Customer Feedback	Cohort-based Metrics
Product Owner	Entrepreneur
CSM (Certified Scrum Master)	CSM (Customer Success Manager)





# lean

software development

## Thank You!

*More Information: [www.poppendieck.com](http://www.poppendieck.com)*